Capacity is the Wrong Paradigm*

Ira S. Moskowitz
Center for High Assurance
Computer Systems-5540
Naval Research Laboratory
Washington, DC 20375

LiWu Chang
Center for High Assurance
Computer Systems-5540
Naval Research Laboratory
Washington, DC 20375

Richard E. Newman
CISE Department
University of Florida
Gainesville, FL
32611-6120

ABSTRACT

At present, "capacity" is the prevailing paradigm for covert channels. With respect to steganography, however, capacity is at best insufficient, and at worst, is incorrect. In this paper, we propose a new paradigm called "capability" which gauges the effectiveness of a steganographic method. It includes payload carrying ability, detectability, and robustness components. We also discuss the use of zero-error capacity for channel analysis and demonstrate that a JPEG compressed image always has the potential to carry hidden information.

1. INTRODUCTION

Steganography is the art and science of sending a hidden message from Alice to Bob, so that an eavesdropper is not aware that this hidden communication is even occurring [23]. We refer to the communication channel from Alice to Bob that transmits this hidden information as a stego channel (it is also sometimes called a subliminal channel [31, 32, 15], although some use that term in a very restricted sense). Note that the stego channel lies hidden in a communication channel, the cover channel, from Alice to Bob—hence the term stego (or subliminal). The cover channel and stego channel are often of the same "data type," but this is not necessary.[1]

One wishes to determine how much "information" [28] can be sent over a stego channel. This is similar to the related information-theoretic studies of covert channels. Covert channels use the paradigm of *capacity* to measure their information carrying ability. There are two important differences between covert and stego channels.

- When studying covert channels no consideration is given to hiding their existence. In contrast, a stego channel only exists if its existence is hidden.
- No consideration is given to how long a covert channel may transmit data. In fact, the channel is tacitly assumed to transmit "forever." On the other hand, a stego channel's transmission time is limited to the type of cover channel/cover medium that is used. For example, if a message is hidden in an image, then the type and size of the image limits the number of transmissions of the stego channel. Therefore, we cannot assume that word sizes of asymptotically rate-maximizing block codes can approach infinity (as is the case w.r.t. covert channel analysis).

Thus, a stego channel is very different from a covert channel. Therefore, we *must* have a new paradigm, because a stego channel is not a covert channel (in the technical sense, not in the vernacular usage of covert).[2]

This is in part because the new paradigm for stego channels must take *detectability* into account, something that is not generally[3] considered when it comes to covert channels (although perhaps it should be). In general, the more data that are hidden, the easier it is to detect it. This is a distinction that is sometimes "hidden" in the literature. Any study of stego channels that does not incorporate some measure of the detectability of the stego channel is seriously flawed; at best it is incomplete, and at worst it is deceptive.

Also, the new paradigm must take into account the pragmatic aspects based on the number of transmissions that are allowed[4] and the effect this has on the ability to devise a code that achieves the theoretical capacity of the channel. Thus, a paradigm other than capacity must be used as

*US Government work. Research supported by the Office of Naval Research.

[1]Note that Prime Minister Thatcher caught leaks from those among her ministers by giving them documents with different word spacing [1], thus the stego and cover channels were very different in form.

[2]In fact, we must pause to ask if capacity is the correct paradigm for covert channels. This question is beyond the scope of this paper; however, it has been touched upon earlier [14]. Either way stego channels and covert channels must be measured differently.

[3]To some extent, it is considered for purposes of auditability of covert channels [35].

[4]We note that in an earlier paper that we presented at NSPW 2000 [15] we discussed a different new paradigm concerning steganography. The concern of that new paradigm was "when is something discovered." We feel that both "new" paradigms are needed for a complete analysis of steganographic systems, and that the two new paradigms are very different.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Capacity is the Wrong Paradigm	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Center for High Assurance Computer Systems,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**13**	

the true metric of a stego channel. The capacity of a communications channel has a specific meaning as put forth by Shannon [28]—it is the upper limit on essentially error-free communication. Theoretically, codes exist that let us send information at any rate less than the capacity, with ϵ-error rate. Attempts to send information at a rate higher than the capacity will result in errors.

Thus, if we simply view a stego channel as a communications channel then we could use capacity as a metric of the stego channel's information carrying potential. However, this totally begs the question of the stego channel's steganographic detectability. Also, it ignores the lifetime of the stego channel. This is why we use a new term — *capability* — when discussing how much information a stego channel can transmit.

Capability is the new paradigm that we propose for stego channels. Capability = (P, D) where P is the payload size and D is a detectability threshold. We sometimes expand the capability to a triple (P, D, R) where R is a measure of robustness of the stego channel. Note that P is a function of the type of coding needed to send the hidden information. For simplicity we restrict ourselves to still image steganography in this paper, but our new paradigm applies across different media. Also, we repeat some examples (and briefly some discussions) that we used in a previous NSPW paper [15], since those examples are best for representing certain concepts. The two papers though are quite distinct.

One must remember that much of what we discuss deals with the semantics of what we were attempting to hide. In extnded work one should consider the implications of the work of Chaitin and Kolmogorov on algorithmic complexity [5]. We have also concentrated on screen images in this paper and have not considered printing issues. Also, progressive type issues concerned with how an image "loads" are not addressed in this paper.

Figure 1: Cover image

Figure 2: Candidate hidden information

Figure 3: Embedded image

Figure 4: Stego image

2. SIMPLE EXAMPLES

This section will explore a few scenarios differentiated by their assumptions about the cover image (greyscale or color), noise (present or absent, correlated or uncorrelated), coding of embedded data (error correction used or not) and embedded data content (quality requirements of the contents to be of use - image or bitstring). All examples use the popular method of image steganography first reported by Kurak and McHugh [10, 15] or a variant of it. This approach hides an embedded image in a cover image by replacing some of the least significant bits (LSB) of the cover image with some of the most significant bits (MSB) of the embedded image. We will refer to this approach as the n-bit KM (n-KM) method when the n LSBs (n-LSB) of the cover are replaced with the n MSBs (n-MSB) of the embedded image. A variant of the n-KM approach is the n-LSB encoding, which simply embeds an arbitrary bitstring in the lowest n bits of an image.

Figure 5: Extracted image (no noise)

Figure 6: Extracted image, $p = .2$

Several images will be used to illustrate the simple examples. Fig. 1 is the cover image. This is the image in which we will do the hiding. Ideally, what we send out (the stego image) of the stego channel should be indistinguishable from the cover image. Fig. 2 represents what we would like to send. Since, for now at least, we are not interested in a 100% true rendition of Fig. 2, we refer to it as the *candidate hidden information* for lack of a better term. Fig. 3 is what we actually hide; it is the same as Fig. 2, except that we use only the MSB of each pixel (brightness) byte instead of all eight bits. Of course we have made an *a priori* decision that this MSB representation of Fig. 2 suffices for our needs. Fig. 4 is the resulting (via Ex. 1) stego image. Fig. 5 is the extracted image if there is no noise (via Ex. 1), whereas Fig. 6 is the extracted image with noise as given in Ex. 2, with $p = .2$.

2.1 Example 1a - Greyscale, 1-KM, no noise, no coding, embedded image

Assume we have greyscale images with dimensions $M \times N$ pixels. Each pixel has a corresponding brightness byte (brightness ranges from 0 to 255). We do not hide an entire image (Fig. 2), but only the MSB bit representation of the image (Fig. 3). This is good enough, unless our concerns are of a more "artistic" nature. This distinction is something that we wish to discuss with the NSPW participants. Using the 1-KM method on Figures 1 and 2 produces Fig. 4. To extract the embedded image, shift every pixel byte of the stego image (Fig. 4) left by 7 bits (Fig. 5).

As a communication channel, this stego channel is noiseless and has a capacity of MN bits per image, or equivalently, 1 bit per pixel. Since there is no noise in this channel, the capacity actually measures how much data can be sent without any error correcting coding being used. Note that this steganography usually[5] cannot be detected by the naked eye (Human Visual System — HVS).[6] We have not yet discussed the degree to which this stego channel is "subliminal." In fact, this stego channel is trivial to detect, so even though it seems as if it can send a great deal of information, the "capability" of this stego channel must be tempered by the fact that it is not very well hidden. Therefore, when making comparisons between stego channels there is more to take into account aside from how many bits can be sent through the stego channel.

2.2 Example 1b - Greyscale, 1-LSB embedding, no noise, no coding, embedded bitstring

We need not limit the embedded message to an actual image, the only thing that matters is how the bits are interpreted. Therefore, we can send any message up to size MN bits via the method described in Ex. 1a. The only limitation to the size of the embedded message is the size of the cover image.

2.3 Example 2a - Greyscale, 1-KM, noise, no coding, embedded image

Now take the same situation as Ex. 1 except that the stego image (the cover image after the embedded image has been "inserted") is subject to random noise. Any bit can be flipped independently with probability p (this is the bit error rate, or BER). Thus, the noise affects each pixel, and each bit in a pixel byte, independently. If we wish to send an embedded image as in Ex. 1a, we can extract a passable representation (Fig. 6) of the embedded image provided p is small.

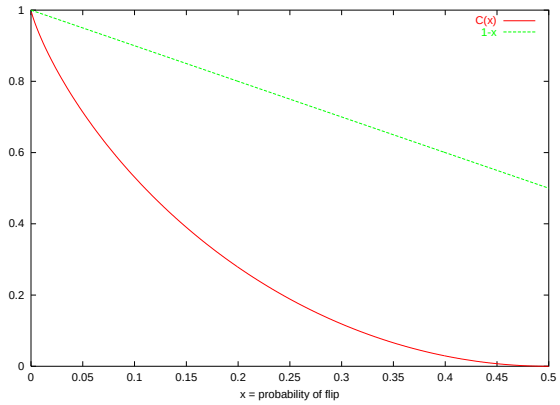

Figure 7: Capacity and 1–BER, plotted against BER

2.4 Example 2b - Greyscale, 1-KM, noise, coding, embedded bitstring

If we view this method of steganography solely in terms of communications theory we see that we have a binary symmetric channel (BSC) which has a capacity of

$$C_{BSC} = 1 - H(p, q),$$

[5] Exceptions to this will be noted later.

[6] Image based steganography cannot be called steganography unless it passes at least the HVS test. This is also a topic that we wish to discuss with the workshop participants.

where $q = 1-p$ and (with all logarithms base 2 throughout),

$$H(p, q) = -(p \cdot \log p + q \cdot \log q).$$

However, we cannot assume that we have infinite uses of this channel; rather, we are limited to MN uses of this channel. Since error correcting codes must be used to obtain a data rate near C_{BSC}, we cannot simply say we can send $MN \cdot C_{BSC}$ bits per image (or C_{BSC} bits per pixel since we are only using the LSB of a pixel byte). This is important — even if detectability is taken into account we see that capacity alone is not the correct measure of how much hidden information we may send via a stego image. Only if the stego channel is "noiseless," as is the case in Ex. 1, does capacity really measure how many bits we can send.

Fig. 7 shows plots of C_{BSC} and the complement of the bit error rate (probability of a bit not flipping), vs. the probability of a bit error (we only plot from 0 to .5, since the capacity is symmetric about .5).

2.5 Discussion of Simple Examples

We must take into account how many bits are truly needed to send the hidden information in a useful manner. In Fig. 6 we can still make out the image of the buildings and important information about their location. Keep in mind that Fig. 6 only has 80% correctness, yet for most needs it contains as much content as Fig. 3. In fact, Fig. 3 has, for many purposes, the same content as Fig. 2. Yet, Fig 3. has 1/8th the number of bits of Fig. 2. This brings us to a deeper problem of *what hidden information are we truly trying to send, and how many bits are needed to represent this information*? (Similar thinking about how "big" a secret is can be found in [17].) When dealing with covert channels and capacity, the conventional wisdom was to consider only "how many" bits we can send and not to concern ourselves with the "nature" of the bits. However, we see that with stego channels we may be willing to accept "noisy" bits as long as the essence of the message is received. This acceptance of noisy bits allows us to decouple the coding problem from the number of bits sent. However, this must be noted in order to compare fairly the steganographic capabilities of different stego channels.

Referring again to Figure 7, consider $p = .2$, where the capacity is .28 bits per pixel. For an $M \times N$ size image we would expect to be able to pass no more than $.28 \cdot MN$ bits. However, we see that this is arguable. In fact, since $p = .2$, we see that $.8 \cdot MN$ bits go through, on the average, without error. This makes sense if we recall that Shannon showed that if you transmit at a rate higher than capacity then you will have errors. One may argue that the information that we are attempting to pass through the stego channel is not really a 1 bit per pixel representation of the embedded image. This is a valid argument. How much information is truly needed to pass the salient parts of the embedded image? Also when we are dealing with an image the HVS is very forgiving when it comes to correcting the erroneous pixels. However, what if the embedded information were not an image, but simply a bit string? Then we could not accept an average error rate of 20% without some sort of correction. In this case the effective rate of .28 bits per pixel, given by the capacity, seems "more" correct.

As mentioned above though, we have not discussed the code needed to send bits at rates approaching the capacity. Pragmatic coding concerns might force us to send far less than $.28 \cdot MN$ bits per image. Therefore, just being able to calculate capacity does not mean that you can transmit in an essentially error-free rate near capacity without doing anything else. You must know the coding with which you are transmitting. Also, the BSC is a trivial channel. Noise characteristics of a channel can be much more complicated (as they are when we discuss AWGN channels later in the paper). It is also possible that the channel is not memoryless. In that situation very little can be said about efficient coding. Keep in mind that we have not yet discussed detectability of the stego channel.

This is why we need a better metric such as *capability*, that incorporates detectability along with the amount and type of information steganographically transmitted.

In the next section, we will embed an image in a second image in such a manner that the extracted image consists only of "noise" and is of no use for steganographic communication in this form. However, the capacity of this stego channel is not zero, and if we concern ourselves with sending bits (which is the proper consideration anyway), and not the "image," we see that the resulting stego channel may in fact pass meaningful information.

3. NOISY COLOR EXAMPLES

We will now use color images. As in the previous greyscale cases, we assume that our images are stored in a lossless manner (e.g., TIFF or BMP). A typical color image has 3 bytes for each pixel: a red byte R, green byte G, and a blue byte B. This results in a 24-bit color image. The color bytes represent the brightness (or intensity) values for each color. A color image can be transferred to a greyscale by using the following formula [20]:

$$Y = .3R + .6G + .1B$$

where Y is the luminance value corresponding to the one brightness byte in the greyscale image, and R, G, and B are the respective integers values of the red, green, and blue bytes in the color image. (Note not all image processing systems are identical. In fact, the software we use, "xv" [37], uses the luminance formula $Y = .3R + .59G + .11B$.) The reason that Y is not simply the average of R, G, and B is that the HVS perceives different colors differently. In fact, the HVS perceives green much more readily than blue (as evidenced by the luminance formula).

We will first discuss our example, and then for the sake of clarity of exposition, describe the important motivation behind it. We now have noise affecting the lower bits of an image, across all three colors. The noise may be independent across R, G, and B, or there may be a dependence across the colors. We will just concentrate of the LSB. Consider the image in Fig. 8, which contains the content that we wish to hide, and the 1-MSB representation of that content as shown in Fig. 9.

Figure 8: Candidate hidden information

Figure 9: Embedded image

Figure 10: Color independent $p = .2$

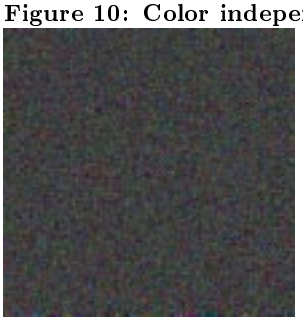

Figure 11: Color independent $p = .5$

By now we hope the reader accepts the fact that we may replace the 1-LSB of a suitable cover image with the 1-MSB image that we wish to embed so that the HVS cannot detect the hiding. Thus, we form the stego image again using the 1-KM method in our color image. In the cases considered below, the stego image may be subject to noise (perhaps due to lossy compression upon saving the stego image in a certain format).

3.1 Example 3.1: Color, 1-KM, color-independent noise, no coding, embedded image

This subsection assumes that the noise affects the LSB of the R, G, and B bytes independently, and is also independent pixel to pixel. We show the resulting extracted image under two different noise conditions. Figs. 10 & 11 are the extracted images (stego image with each byte shifted seven places to the left). Fig. 10 is the result of subjecting the embedded image (Fig. 9) to a noise that inverts each bit with probability $p = .20$, independently across R, G, and B. Fig. 11 is the results of flipping each color bit independently with probability $p = .50$. Fig. 10 still has meaningful content, whereas Fig. 11 is just random noise and has no content. The reason that Fig. 11 is random noise is that each three bit pair representing a pixel in Fig. 9 has an equi-probable chance of becoming any three bit pair. For example the pixel which has a LSB of (1,0,0) has a 1/8 probability of the LSB transitioning into any of $\{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0),(0,1,1),(1,0,1),(1,1,1)\}$. Fig. 11 is the result of this experiment.[7]

[7]Let us review our representation. The embedded image is the LSB plane of the stego image, written as (x_1, x_2, x_3), where x_1 is the R value, x_2 is the G value, and x_3 is the B value. Therefore, per pixel of the stego image, the embedded image is given as (x_1, x_2, x_3), where $x_i = 0$ or 1. However since this is really the MSB of the image we are hiding (x_1, x_2, x_3) is interpreted as $R = x_1 \cdot 128$, $G = x_2 \cdot 128$, and $B = x_3 \cdot 128$ w.r.t. the extracted image.

We now consider the range of p between 0 and .50 (we do not concern ourselves with $p > .5$ because that just results in "negative" images, and the capacity of the associated channels are identical for p and $1 - p$). In terms of a communication channel we have an input alphabet of size eight. The input alphabet is

$$\{(0,0,0),(1,0,0),(0,1,0),(0,0,1),$$
$$(1,1,0),(1,0,1),(0,1,1),(1,1,1)\} .$$

Since bits are flipped independently across the triples the output alphabet is the same as the input alphabet. Let us consider the input symbol $x_1 = (0,0,0)$. The symbol x_1 may not be changed at all and result in output symbol $y_1 = (0,0,0)$ with probability $(1-p)^3$; x_1 can be changed to output symbols $y_2 = (1,0,0)$, $y_3 = (0,1,0)$, or $y_4 = (0,0,1)$, each with probability $p(1-p)^2$; or x_1 can be changed to output symbols $y_5 = (1,1,0)$, $y_6 = (0,1,1)$, or $y_7 = (1,0,1)$, each with probability $p^2(1-p)$, or with probability p^3 to output symbol $y_8 = (1,1,1)$. The other input symbols behave similarly.

Consider finite discrete random variables A and B, $a_j \in A$, $b_i \in B$. The entropy of B, $H(B)$, is:

$$H(B) = -\sum_{i=1}^{n_B} p(b_i) \log p(b_i) .$$

We define the conditional entropy (equivocation), $H(A|B)$, as:

$$H(A|B) = -\sum_{i=1}^{n_B} p(b_i) \sum_{j=1}^{n_A} p(a_j|b_i) \log p(a_j|b_i) ,$$

where n_A (n_B) is the number of non-probabilistically trivial values of A (B). (Values whose probability is zero do not affect the terms of interest.)

Given a discrete memoryless channel (DMC) the output symbols y_j are the values of the output random variable

Y, and the input symbols x_i are the values of the input random variable X. The channel matrix $[p(y_j|x_i)]$, where $p(y_j|x_i)$ is the conditional probability of the output symbol y_j given that the input was x_i is [8]

$$[p(y_j|x_i)] = \begin{array}{c} \\ x_1 \\ \vdots \\ x_{n_X} \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & \cdots & y_{n_Y} \end{array} \\ \begin{pmatrix} p(y_1|x_1) & \cdots & p(y_{n_Y}|x_1) \\ \vdots & \cdots & \vdots \\ p(y_1|x_{n_X}) & \cdots & p(y_{n_Y}|x_{n_X}) \end{pmatrix} \end{array} .$$

For a DMC the channel matrix completely describes the channel and the capacity C [28] is given by maximizing the mutual information $I(X,Y)$,

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \ ,$$

over the distributions that support $\{x_i\}$ (the symbols x_i are fixed, the probability values $p(x_i)$ vary), so

$$C = \max_X I(X,Y) \ .$$

We say that a channel is a *symmetric* DMC, or more simply put a *symmetric channel*, if the channel is a DMC and every row of the channel matrix is the same up to permutation, and every column of the channel matrix is the same up to permutation [3]. For a symmetric channel, $\sum_j p(y_j|x_i) \log p(y_j|x_i)$ is independent of i since the rows of a symmetric channel matrix are the same up to permutation. Therefore, without loss of generality, $H(Y|X) = -\sum_j p(y_j|x_1) \log p(y_j|x_1)$. So maximizing $I(X,Y)$, over different distributions of X, comes down to maximizing $H(Y) \leq \log n_Y$. If we can show that there exists a distribution of X such that $H(Y) = \log n_Y$, we will know the maximum of $H(Y)$. Let X have the equiprobable distribution $p(x_i) = (1/n_X)$ for all i. Since $p(y_j) = \sum_i p(y_j, x_i) = \sum_i p(y_j|x_i)p(x_i) = (1/n_X)\sum_i p(y_j|x_i)$, the term $\sum_i p(y_j|x_i)$ is the same for all j because this is the sum of the jth column entries, which are the same for all j. Therefore $p(y_j)$ is independent of j, so Y has the equiprobable distribution $p(y_j) = 1/n_Y$ when X has the equiprobable distribution. Hence, it is possible that $H(Y) = \log n_Y$, so we have determined the maximum mutual information $I(X,Y)$, and the following is the capacity of a symmetric channel:

$$C = \log n_Y + \sum_j p(y_j|x_1) \log p(y_j|x_1) \ . \qquad (1)$$

For the color-independent noise case, the channel matrix is an 8×8 matrix with every row and column, up to permutation, of the form $\{q^3, pq^2, pq^2, pq^2, p^2q, p^2q, p^2q, p^3\}$, where $q = 1 - p$:

Of course when $p = 0$, this results in the 8×8 identity matrix, and $p = q = 1/2$ results in the 8×8 matrix where every entry is $1/8$. Regardless of the p value, every row has the same entries (up to permutation), and every column has the same entries (up to permutation). Thus our channel is a symmetric channel. By Eq. (1), the capacity C of this channel is

$$C = 3 + (3p^3 + 6p^2q + 3pq^2)\log p + (3q^3 + 6pq^2 + 3p^2q)\log q \ . \qquad (2)$$

[8] We annotate rows and columns of matrices for clarity.

Therefore the capacity is $0 \leq C \leq 3$ bits per symbol, as p varies from .50 down to 0, and C achieves the boundary values of 0 and 3, respectively.

3.2 Example 3.2: Color, 1-KM, color-dependent noise, coding, embedded bitstring

We now assume that the noise is still pixel-wise independent, but it is totally dependent across R, G, and B. In other words the LSBs for each color either all change simultaneously or none of them change. Observe what happens to the embedded image Fig. 9 under such noise effects.

Figure 12: Color dependent $p = .2$

Figure 13: Color dependent $p = .5$

The extracted image in the $p = .5$ case (Fig. 13) is not random noise, because there are still some residuals from the embedded image in it. However, in terms of an image, it is essentially useless. Recall that with color-independent noise, Fig. 11 is random noise, while there are still some residual elements of Fig. 9 in the color-dependent Fig. 13. This is because color-dependent noise behaves differently from color-independent noise. Given a three bit representation of the LSB of a pixel (b_1, b_2, b_3), we define the complement of that three bit representation to be the three bit tuple (b_1^c, b_2^c, b_3^c), such that the term by term exclusive-or of (b_1, b_2, b_3) with (b_1^c, b_2^c, b_1^3) is $(1, 1, 1)$. With this in mind we study what may happen to the MSB representation of an image under color-dependent noise. A region that is very dark (or very bright) transitions to a region that is a mix of very dark and very bright. However, a region that is very bright with respect to one color transitions to a region that still has this one color mixed with the "complementary" color. This behavior is seen in Fig. 13.

So, some of the information about the image is still able to be extracted even when $p = .50$, in contrast to the color-independent noise situation where no part of the image is

$$
\begin{array}{c} \\ (0,0,0) \\ (1,0,0) \\ (0,1,0) \\ (0,0,1) \\ (1,1,0) \\ (1,0,1) \\ (0,1,1) \\ (1,1,1) \end{array}
\begin{array}{c}
\begin{array}{cccccccc} (0,0,0) & (1,0,0) & (0,1,0) & (0,0,1) & (1,1,0) & (1,0,1) & (0,1,1) & (1,1,1) \end{array} \\
\left(\begin{array}{cccccccc}
q^3 & pq^2 & pq^2 & pq^2 & p^2q & p^2q & p^2q & p^3 \\
pq^2 & q^3 & p^2q & p^2q & pq^2 & pq^2 & p^3 & p^2q \\
pq^2 & p^2q & q^3 & p^2q & pq^2 & p^3 & pq^2 & p^2q \\
pq^2 & p^2q & p^2q & q^3 & p^3 & pq^2 & pq^2 & p^2q \\
p^2q & pq^2 & pq^2 & p^3 & q^3 & p^2q & p^2q & pq^2 \\
p^2q & pq^2 & p^3 & pq^2 & p^2q & q^3 & p^2q & pq^2 \\
p^2q & p^3 & pq^2 & pq^2 & p^2q & p^2q & q^3 & pq^2 \\
p^3 & p^2q & p^2q & p^2q & pq^2 & pq^2 & pq^2 & q^3
\end{array} \right)
\end{array}
.
$$

channel matrix: color independent case

extracted. We studied the underlying communication channel in the color-independent case and saw that the capacity is zero when $p = .50$. How does the communication channel behave when we have color-dependent noise? The input alphabet and output alphabet are the same as for the color independent noise (see subsection 3.1). What is very different is the channel matrix.

Consider the input $x_1 = (0, 0, 0)$. Since the noise is color dependent, $(0, 0, 0)$ either stays as $(0, 0, 0)$ with probability q, where $q = 1 - p$, or it is transformed to $(1, 1, 1)$ with probability p. Note that the input symbol $(1, 1, 1)$ either stays as $(1, 1, 1)$, or is transformed to $(0, 0, 0)$.

Since this is a symmetric channel, by Eq. (1) the capacity is

$$C = 3 + p \log p + (1 - p) \log(1 - p) \; . \tag{3}$$

What is very interesting about Eq. (3) is that the capacity is always bounded from below by 2, $2 \leq C \leq 3$. In fact, we see that pairs of input symbols map to pairs of output symbols reflexively in pairs. In other words:

- $\{(0, 0, 0), (1, 1, 1)\} \rightarrow \{(0, 0, 0), (1, 1, 1)\}$
- $\{(1, 0, 0), (0, 1, 1)\} \rightarrow \{(1, 0, 0), (0, 1, 1)\}$
- $\{(0, 1, 0), (1, 0, 1)\} \rightarrow \{(0, 1, 0), (1, 0, 1)\}$
- $\{(0, 0, 1), (1, 1, 0)\} \rightarrow \{(0, 0, 1), (1, 1, 0)\}$.

Therefore if we view the four pairs above as equivalence classes we can form a secondary channel which has the 4×4 identity matrix for the channel matrix. Therefore, no matter what p is, we can always send 2 bits of information. In fact, there is no noise affecting this secondary channel so the $C = 2$ is always achievable without any coding! (Note that the actual channel has $C > 2$ for $0 < p < 1/2$ (as in the other example this channel is symmetric about $1/2$), but coding is required to achieve this data rate. Given that our channel is actually a stego channel we might not have "enough transmissions" to utilize a coding that approaches capacity.) This leads us to the concept of *zero error capacity* denoted by C_0 [29]. Of course we require no error correction to achieve the zero error capacity in the situation we have shown. This may not always be the case, though.

3.3 Dependent or Independent?

In the above examples we see that when there is a total dependence among the color bytes with respect to noise, that information may still be passed, even in the noisiest of situations. However, for color-independent noise, it is possible for no information to be passed. If we are dealing with JPEG, the true answer lies somewhere in between. This is because JPEG operates not in the RGB coordinate system, but rather in the YUV coordinate system. We know from the above formula that Y is the luminance of the pixel. U and V are chrominance values. U is the difference between R and Y, whereas V is the difference between B and Y. What is important is that the YUV coordinate systems expresses a dependence between the colors. This dependence translates to a dependence of the noise between the colors R, G, and B when an image is saved as a JPEG. Thus, we conjecture that *even the most severely compressed JPEG image may pass some hidden information in the LSBs.* This is a very strong statement and may give a theoretical existence proof of robust (survives attacks from compression noise) steganography with respect to JPEG images. We will discuss this in future work.

4. PARTIAL SUMMARY

The above examples and discussions are worth summarizing.

- How much information are we truly hiding? The important parts of an image might be describable by a relatively small number of bits. Therefore it might be better to speak of "embedding information" rather than to speak of "embedding an image." We believe that this distinction is sometimes glossed over in "popular" discussions of steganography (most technical papers correctly discuss embedding files). Of course, considering the embedded information as an image has the advantage that the HVS can correctly parse through errors via the implicit semantics of an image. Thus, an image file is a very special file, one can easily get through the errors in it, whereas in another file type error-correction may be necessary to send any information (of course we are implicitly using an image "viewer"). Audio files might behave in a manner similar to image files, but an arbitrary bitstream cannot recover so gracefully from errors. Depending upon the coding difficulties it is perhaps better to speak about how many bits a stego channel may transmit, rather than how big an image it can transmit.

$$
\begin{array}{c} \\ (0,0,0) \\ (1,0,0) \\ (0,1,0) \\ (0,0,1) \\ (1,1,0) \\ (1,0,1) \\ (0,1,1) \\ (1,1,1) \end{array}
\begin{array}{c}
\begin{array}{cccccccc} (0,0,0) & (1,0,0) & (0,1,0) & (0,0,1) & (1,1,0) & (1,0,1) & (0,1,1) & (1,1,1) \end{array} \\
\left(\begin{array}{cccccccc}
q & 0 & 0 & 0 & 0 & 0 & 0 & p \\
0 & q & 0 & 0 & 0 & 0 & p & 0 \\
0 & 0 & q & 0 & 0 & p & 0 & 0 \\
0 & 0 & 0 & q & p & 0 & 0 & 0 \\
0 & 0 & 0 & p & q & 0 & 0 & 0 \\
0 & 0 & p & 0 & 0 & q & 0 & 0 \\
0 & p & 0 & 0 & 0 & 0 & q & 0 \\
p & 0 & 0 & 0 & 0 & 0 & 0 & q
\end{array} \right)
\end{array}.
$$

channel matrix: color dependent case

- Considering the stego channel simply as a communication channel is the wrong approach. We must not forget that block codes need to be designed to achieve rate near capacity and that the stego channel might not be able to transmit a sufficient number of times (think number of pixels), so that the code can effectively transmit information. On the other hand, there might be codes that are not very large and therefore do not require a large number of transmissions to achieve a sub-optimal rate of transmission. This sub-optimal rate might be more than sufficient to send effective amounts of hidden information. The zero-error capacity discussed above involves such a code.[9] For the example we showed that "no" coding (to be precise, the identity coding) suffices.

As noted earlier, even with respect to covert channels the sole use of capacity has been called into question, e.g., the *small message criterion* [14]. The reasoning behind the small message criterion is not directly applicable to stego channels, because we do not have the luxury of infinitely many transmissions. But the idea that measures other than capacity are useful still holds. Note that in the related information hiding field of watermarking, Sugihara [33] expressed concern with simplistic applications of Shannon's capacity as the sole measure of embedded information transfers.

5. DISCUSSION

So far, we have just been focusing on how much information a stego channel may send. Remember a stego channel in general becomes useless if its existence is made known. Some qualifications to this are noted below, however, it is certainly the case that if it is possible to detect whether a steganographic channel is being used, that it is no longer fulfilling its purpose. Therefore, for stego channels we must include a measure of detectability when discussing their usefulness.

5.1 Ex. 1 Revisited

Let us revisit Ex. 1, the 1-KM method. Using the 1-KM method, we can transmit MN bits per image. In terms of a communication channel, this is a noiseless DMC with a capacity of 1 bit per pixel. Of course, there remains the caveat that we are limited to MN transmissions. The 1-KM method cannot be discovered by the HVS (for most cover images). If a proposed method of steganography is not detectable by the HVS is that good enough? The answer is a resounding no!

How detectable is the 1-KM method? Well if we know the algorithm all that is required is to take the suspect image and shift the bits 7 to the left. If you see a different image there, the game is over! In general, detection tools that do not involve human interpretation are preferable.

5.2 Detection

There are many techniques for detecting steganography (i.e., steganalysis [7, 24, 25]). In fact, many often take the Kerckhoffs approach that is applied to cryptography [2] — assume that the method of steganography is known, yet the use of steganography should still not be detectable without the key. As discusssed later in this paper, a weaker condition may suffice. Regardless, the tradeoff between capacity or payload and detectability requires further investigation.

The detection tool just discussed rests upon interpreting the 1-LSBs as a hidden image. An alternative approach would be to run various statistical tests. One such test is the discrete Laplacian[10] $\nabla(p_{x,y})$, where $p_{x,y}$ is the (x,y) pixel. $\nabla(p_{x,y})$ works by measuring the difference in local pixel neighborhoods.

$$\nabla(p_{x,y}) = p_{x+1,y} + p_{x-1,y} + p_{x,y+1} + p_{x,y-1} - 4p_{x,y}$$

$\nabla(p_{x,y})$ is not defined for boundary pixels. That is, for an $M \times N$ image, $\nabla(p_{x,y})$ is not defined for $x = 0$ or for $y = 0$, nor for $x = M - 1$ or $y = N - 1$. (Keep in mind that a $M \times N$ image is interpreted as a $M \times N$ matrix. However the indexing goes left to right, from 0 to $M - 1$, in the horizontal direction, and top to bottom, from 0 to $N - 1$, in the vertical direction.)

Let us look at (the midrange of) the histogram of the discrete Laplacian of a legitimate TIFF image (Fig. 14), and the same range of the discrete Laplacian of a 1-KM stego image (Fig. 15). Fig. 14 is the discrete Laplacian of the cover image Fig. 1, whereas Fig. 15 is the discrete Laplacian of the stego image Fig. 4. The graphs of very different: the discrete Laplacian of the stego image shows humps every 2 values. This is because the 1-LSBs have been affected. The 1-LSBs of a legitimate image are not as correlated as the 1-MSBs of a legitimate image. Therefore when we replace the

[9] Codes for achieving zero-error capacity are not that well studied [29].

[10] The use of the discrete Laplacian as a detection tool was briefly discussed at NSPW but not published. A discussion of it may also be found in Katzenbeisser and Petitcolas [8]

1-LSBs of the cover image with the 1-MSBs of the embedded image, under the KM approach, we see that the 1-LSBs of the resulting stego image have the *wrong* statistical signature. This is shown by the humps in Fig. 15.[11] A tool such as this could be automated to look for incorrect LSB signatures, whereas machine interpretation of some bit planes [7] as part of an image is a more difficult problem related to the field of artificial intelligence and computer vision.

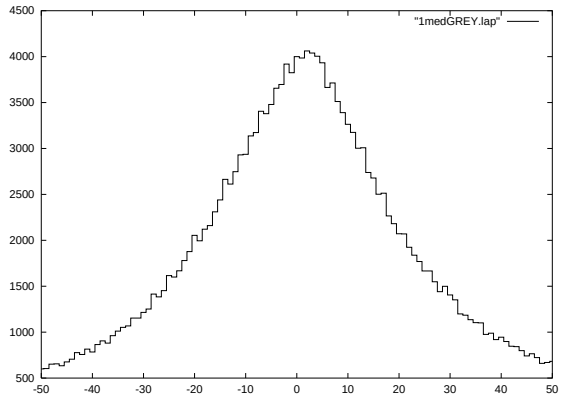

Figure 14: Cover image discrete Laplacian

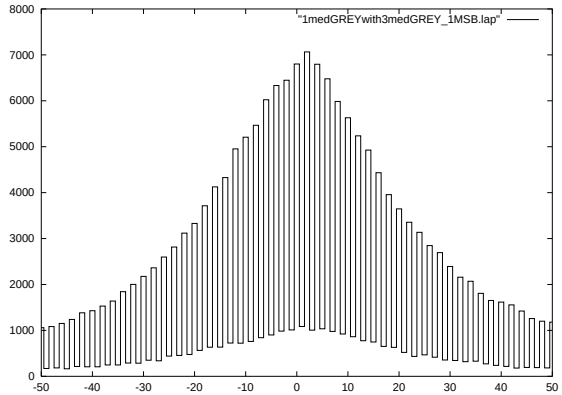

Figure 15: Stego image discrete Laplacian

What if we attempt to introduce noise (e.g., by encrypting the embedded data) into the LSBs under the KM approach, is the steganography still visible? The answer is yes, the histogram no longer has the humpy behavior indicative of LSB hiding, but the the histogram has greater variance. At this time we have no hard and fast rules. Therefore, even though the discrete Laplacian no longer shows the telltale humpy behavior of bit plane replacement, we still see that the discrete Laplacian may still reveal some information. However, the detection has now become more difficult.

We note though that all bit planes of an image seem to have certain dependencies, especially in the bright areas. This is especially true of images that originated as JPEGs. (The comments in this subsection are not backed by enough experimentation or theory. However, we feel that they are on the correct path.) Therefore, if the LSBs have been encrypted to appear as random noise (and lessen any detection that the discrete Laplacian may show), other tests may detect that something is wrong with the LSBs. This is new territory and ripe for discovery.

[11] One need not restrict themselves to just the LSBs; the detection works similarly for the n-LSBs.


Figure 16: 24 bit color image

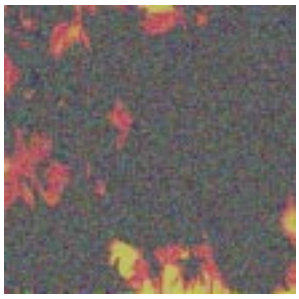
Figure 17: 2-LSBs, shifted left 6 bits

Embedding data in a cover image generally introduces artifacts, which constitute the basis for detection. One form of artifact is apparent when we consider the TIFF file shown in Fig. 16. Fig. 17 is the 2-LSB of that TIFF file, with every byte shifted six places to the left. We see that the bright areas of Fig. 16 work their way down to the lower bits. Fridrich has noted similar behavior [6], as have Lee and Chen [11]. Steganography that does not respect such "artifacts" is detectable, or at least highly suspicious.

NRL [15] has modified the KM approach to only hide a small message in a lossless manner. We have experimental evidence that our method is essentially impossible to detect [16]. Of course, this is with the present detection tools. Perhaps in the future someone will determine a way to easily detect the NRL method. Therefore, in general, any measure of undetectability may vary over time.

In any case, when discussing steganography and the capacity, data rate, or capability of the associated stego channels, we must include a measure of detection.

5.3 Robustness

One may also want to take robustness of the steganography into account. If we can hide a message that survives JPEG compression we have come up with a very strong method. If a steganographic method must be restricted to compressionless formats, we could (for example) eliminate all possibility of steganography on a web site by forcing all the images to be stored as JPEGs instead of TIFFs. This may obviate the need to detect stegoimages reliably, if the goal is merely to

prevent their use.

It is also the case that error correction coding needed to overcome impairments on the stego channel may itself increase detectability. Error correction coding necessarily introduces redundancy into the embedded data by its very nature. It is likely that this redundancy can be exploited by detection mechanisms; this is the case whenever error correction coding is used, regardless of whether encryption (or compression) is used in the system.

Except for transposition ciphers, encryption that is performed on the source embedded data to randomize them and to prevent their disclosure must be done before the embedded data are error correction coded. This is because, at the receiving end, the errors must be removed before decryption is performed. Any cryptosystem that has dependencies of many of the plaintext bits on many of the ciphertext bits (i.e., has good diffusion) will fail to function if there are errors in the ciphertext. Thus, error correction decoding must be performed before decryption in order to remove all errors to obtain accurately decrypted data. Use of transposition ciphers (i.e., permuting the data) after error correction coding may be of some use for confusion purposes, but it is very limited with regard to how much it can change the characteristics of the data, or the degree to which this can prevent detection.

6. OUR NEW PARADIGM

Based upon our discussions we see that a stego channel may be measured by a tuple

$$\text{Capability} = (P, D)$$

referred to as the capability. This is the formalization of our new paradigm. P is the payload, which is the amount and type of information that actually can be sent, through realistic and pragmatic coding, with the threshold of detection kept under D.

6.1 Payload

In general, if no data type is given for the embedded information, we assume that it is simply a bit string (in other words, unless noted otherwise, we would not be concerned with sending an image, but rather concern ourselves with the bits that could express the image—see the discussion in section 1.) If the payload is concerned with something other that a bit string, say an image, then we may include a fidelity factor with P. Assuming that the embedded message is a bit string is the best approach, and, as noted, is the default. This is also the standard approach for dealing with communication channels. The issue of source coding is not taken into account. Data types such as images can lead to confusion and interpretive mistakes. The essence of what we want to send, should be a mathematical construct, not an fuzzy concept subject to interpretation. When discussing the payload in terms of a generic bit string we will use bits/pixel (or bits/image) as the unit (of course we can generalize to cover messages that are not still images and thus change the units). We again emphasize the point that we should concentrate only bit strings, rather then images.

Consider Fig. 1. As a TIFF file it is 250198 bytes, and when we save it to a JPEG (quality factor 100%) it shrinks slightly to a size of 224174 bytes. The TIFF and JPEG are indistinguishable to the HVS. Note that the actual size of the image in Fig. 1 is 176×176 mm. Fig. 18 shows the result of turning Fig. 1 into a thumbnail of size 2917 bytes (reducing from 500x500 to 125x125 pixels and saving in the default JPEG mode of xv). This thumbnail is shown in its actual size of 44x44 mm. Forgetting about image formats, we were interested in the MSB representation of Fig. 1, which is 250000 bits. However, we may be able to represent the essence of that in a file that is only $2917 \times 8 = 23336$ bits. Even better since the thumbnail is 125x125 pixels if we only care about the MSB, then 15625 bits are all that are needed (further attempts to use standard compression tools did not let us reduce the size further). We have not worked on optimizing this, so we take 15625 bits as an upper limit. Thus, we have lowered the "size" by an order of magnitude, but have lost minimal "meaningful information." Therefore, with proper error correcting coding we may be able to send the "essence" of Fig. 1 in a very noisy environment.[12] Again, this is the standard approach to measuring how much "information" can be sent via a transmission scheme.

Figure 18: JPEG thumbnail of Fig. 1

We see that given a noisy transmission we may still be able to send all of the intended message, provided that we use the proper error correction in our coding for transmission over the stego channel. We emphasize that this distinction is often forgotten when it comes to steganography. A legitimate reason for this is that the coding issues can be quite difficult, whereas, sending an image that results in the same image with some degradation (still good enough to get the point across) is easier to do and to explain. However, for a proper analysis of the danger of any stego channels we must explore all aspects of the message payload.

6.2 Detection

The detection factor D is itself not that well-defined. Steganography must not be apparent to the human eye. If it is, then we have not performed steganography in any sense of the word. The idea behind the steganographic communication, at least for an image, is that we cannot tell by looking at an image that there is something hidden in it. Of course, this comes with the caveat that not just any image is used

[12] In order to embed an error-correction coded version of the thumbnail in the cover using 1-KM, a net capacity of about 0.10 is required. It is not unreasonable to assume that half the Shannon capacity can be achieved, so a Shannon capacity of 0.20 should suffice. From Figure 7, this corresponds to a BER of about 0.24 or less.

as a cover image. For example, if we use a cover image such that every pixel is black (e.g. the bytes are zeroed out), then even a few bits hidden in such an image could be detected by the HVS. The concept of what is good enough for a generic cover image has not been put on a firm foundation. But we believe enough has been said to satisfy the reader that a minimum condition of steganography is that it not be visible to the HVS.

Kerckhoffs' principle [2], a standard of cryptography that the "security" of a cryptosystem should hold even if the algorithm is known, i.e., that its security should depend only upon the key, may not apply in all steganographic cases. Obviously if we are given 10 images to examine and are told that a KM method has been used, then it is trivial to detect the steganography. But what if we have to check every image on the Usenet newsgroups, or the entire web? Would knowing that the KM method was used on some of the images allow us to detect the steganography (in a reasonable amount of time)? Of course, the designer of a steganographic system should still aim to satisfy Kerckhoffs' principle, but it might not be necessary in all situations.

What tools do we have to study an image? To do the detection analysis correctly we must state exactly what detection tools are at our disposal. Remember that a stego channel ceases to exist once it has been discovered. In general, when dealing with detection we assume that we have a "good" cover image with which to work. Some methods of steganography are adaptive to the cover image and adjust the hiding to process so as to make it undetectable by the HVS [11, 9, 19]. These concepts should also be discussed when it comes to D.

Also keep in mind that detection need not be done only in the spatial domain (pixels and their R,G,B values). One can transform an image from the spatial to the frequency domain (e.g., descriptions of these techniques are given in [4]). Steganography can be done in the frequency domain. Therefore we should have detection tools for the frequency domain also [6, 24, 25]. Frequency domain approaches give us the ability to embed the message in a manner that is robust to LSB corruption. However, we may detect such attempts by studying the coefficient values of the various frequency transforms, and looking for statistical anomalies [24, 36]. (Note this approach for hiding information works quite well for watermarking, where it does not matter that there is "hidden" information. What matters for watermarking is that the "hidden" information not interfere with the cover image and that the "hidden" information be robust to removal. In short, steganography values detection over robustness, whereas watermarking values robustness over detection.)

Hiding techniques for JPEG images often do their hiding in the frequency domain, e.g. Jsteg [34] and F5 [36]. This is because JPEG converts 8×8 blocks of the spatial domain into a frequency domain by using the discrete cosine transform [30]. Detection of Jsteg is discussed elsewhere [7, 25]. Of course, we need not restrict ourselves only to transforms that arise from JPEG [26, 27]. Note that a recent method of hiding in the spatial domain [18] works against the JPEG-compatibility detection method proposed by Fridrich [6].

Marvel et al. [12, 13] have also done work (in the spatial domain) that treats the cover as noise, and transforms the information to be embedded into Gaussian noise, which is added to the cover. The stego channel is thus modeled so that it is bounded by additive white Gaussian noise (AWGN) channel. The capacity of the AWGN [28] is well-known and based upon the signal to noise ratio of the channel. Note that Marvel's work improves on earlier methods that use the AWGN as the stego channel model. The detectability of this stego channel is based upon the HVS and the signal to noise ratio. We feel that more than the signal to noise ratio is needed to satisfy the undetectability conditions. The signal to noise ratio's size is a necessary, but not sufficient condition. We will explore this concept in future work to see if our claim is true.

6.3 Robustness

One can also extend capability to a triple (P, D, R). The factor R is a measure of the robustness of the steganographic method to noise. If the method only holds for lossless formats this should be noted. If the embedding can stand up to JPEG compression, the type and quality factor of the JPEG method should be noted. If the embedding fails only against attacks that severely degrade the cover image, this too should be noted.

It is sometimes possible to interrupt steganographic communication without the need of detecting the steganographic communication. For example, consider any steganographic method that uses the 2-LSB. If we had the ability to scramble the two lower bit planes then (1) the stego channel would be useless, and (2) the cover image would not loose much visual fidelity. This is a possible method for preventing steganography. This type of approach is similar to the use of Stirmark in destroying the synchronization needed to read a digital watermark [21, 22].

6.4 Examples of Capability

In this section we illustrate our new paradigm by example.

6.4.1 Capability of Example 1

Capability = (P: 1 bit/pixel, no coding necessary.
D: knowledge of the algorithm renders this useless unless an adaptive encryption is used prior to the embedding so that the LSB pattern has the correct artifacts—the discrete Laplacian can reveal embedding, use of encryption can lessen this revelation, but further research is required into the discrete Laplacian and other statistical techniques.
R: not robust—lossy compression can destroy the embedded message.)

6.4.2 Capability of Example 2

Capability = (P: If the noise p is not too large then MSB represented images can be transmitted noisily, but recognizably. In terms of a bit string (bits/pixel) the "capacity" (in the sense of Shannon) is $1 - H(p, 1-p)$ bits/pixel. But, to achieve this rate we must be concerned with the complexity of the coding, and also the world length of the code.
D: If the algorithm is known, this method is trivially detectable if we are sending images (with no encryption). If we are sending a bit stream, them the detection is more subtle, but still not too difficult.

R: This approach incorporates robustness by accounting for noise, so the robustness is "built in." Of course additional or bursty noise can affect all of the stego channel's characteristics.)

6.4.3 Capability of Example 3.1

This is similar to Example 2.

6.4.4 Capability of Example 3.2

Capability = (P: We only concern ourselves with a bitstring. We can send 2 bits/pixel without any error correcting coding, and send $2MN$ bits per $M \times N$ image. If $p < .5$ we may send more than 2 bits/pixel, but more complex coding must be used. Also, we must take the length of the code words into account in order to get a per image payload figure.
D: We are presently studying this for large p. We feel that detection will be difficult in very noisy situations (such as severe lossy compression). Of course, bits should be scrambled before embedding to confuse eavesdroppers. However, with high noise levels, legitimate image artifacts can become lost.
R: This approach survives correlated noise, but not uncorrelated noise (via coding as explained in subsection 3.2). We conjecture that an approach such as this will guarantee a non-zero, hard to detect, method for JPEG compression.)

7. CONCLUSION

Stego channels are not easy to quantify. Their payload size and usefulness come with caveats. The user must be aware of the strengths and weaknesses of the steganographic method in use. Comparisons between stego channels may be impossible to make in certain situations. This paper serves as notice that when dealing with steganography, it may not be business as usual.

Concepts such as zero-error capacity and the ease of coding for a communication channel must be taken into account. One cannot assume that they have infinite transmissions with a stego channel. If each pixel (8×8 block, etc.) is treated as a transmission, then we are limited to the number of pixels (8×8 blocks, etc.) times the number of images.

We propose a new paradigm for measuring how much "stuff" a stego channel can transmit. This new paradigm is a tuple called the capability: it measures how much and what type of information is being sent, it includes a measure of the detectability of the stego channel, and it may include the robustness of the stego channel against attack.

8. ACKNOWLEDGEMENTS

The authors thank the hidden reviewers and the workshop participants for their helpful comments.

9. REFERENCES

[1] R. Anderson. *Stretching the Limits of Steganography,* in R. Anderson (Ed.) Information Hiding, LNCS 1174, IH'96, pp. 39-48, Springer 1996.

[2] R. Anderson. *Security Engineering,* Wiley, 2001.

[3] R.B. Ash. *Information Theory,* Interscience Publishers 1965, republished Dover Publications 1990.

[4] L. Chang. *Issues in Information Hiding Transform Techniques,* NRL Memorandum Report, 2002.

[5] T.M. Cover and J.Y. Thomas. *Elements of Information Theory,* Wiley, 1991

[6] J. Fridrich, M. Goljan, and R. Du. *Steganalysis Based on JPEG Compatibility,* in A. Tescher, B. Vasudev, & V.M. Bove, Jr. (Ed.) Proc. SPIE Vol. 4518 (2001), Multimedia Systems and Applications IV, pp. 275-280.

[7] N.F. Johnson, Z. Duric, and S. Jajodia. *Information Hiding: Steganography and Watermarking—Attacks and Countermeasures,* Advances in Information Security 1, Kluwer Academic Publishers, 2001.

[8] S. Katzenbeisser and F.A.P. Petitcolas (editors). *Information Hiding techniques for steganography and digital watermarking,* Artech House, 2000.

[9] E. Kawaguchi and R.O. Eason. *The principle and applications of bpcs-steganography,* in SPIE International Symposium on Voice, Video, and Data Communications: Multimedia Systems and Applications, pages 464–473, Boston, MA, November 2-4 1998.

[10] C. Kurak & J. McHugh. *A Cautionary Note on Image Downgrading,* in Computer Security Applications Conference, San Antonio, TX, USA, pp. 153-159, Dec. 1992.

[11] Y. Lee and L. Chen. *An adaptive image steganographic model based on minimum-error lsb replacement,* in Ninth National Conference on Information Security, pages 8–15, Taichung, Taiwan, 14-15 May 1999.

[12] L.M. Marvel. *Image Steganography for Hidden Communication,* PhD dissertation, Dept. of Electrical Engineering, Univ. of Delaware, Spring, 1999.

[13] L.M. Marvel, C.G. Boncelet, Jr., and C.T. Retter. *Spread Spectrum Image Steganography,* IEEE Trans. on Image Processing, Vol. 8, No. 8, pp. 1075–1083, August 1999.

[14] I.S. Moskowitz and M.H. Kang. *Covert Channels — Here to Stay?* Proc. COMPASS, Gaithersburg, MD, pp. 235-243, IEEE Press, 1994.

[15] I.S. Moskowitz, G.E. Longdon, and L. Chang. *A New Paradigm Hidden in Steganography,* Proc. NSPW, Ballycotton, County Cork, Ireland, pp. 12-22, ACM, Sept. 2000. Reprinted in "The Privacy Papers," (Ed. R. Herold) pp. 331-349, Auerbach, 2002.

[16] I.S. Moskowitz, N.F. Johnson, and M. Jacobs. *A Detection Study of an NRL Steganographic Method,* NRL Memorandum Report: forthcoming 2002.

[17] R. Nelson. *What is a Secret and What does that have to do with Computer Security?* Proc. NSPW, Rhode Island, pp. 74-79, 1994.

[18] R. E. Newman, I. S. Moskowitz, L. Chang, and M.M. Brahmadesam. *A Steganographic Embedding Undetectable by JPEG Compatibility Steganalysis,* to appear, Proc. Information Hiding Workshop, Oct. 2002, the Netherlands, IH 2002.

[19] M. Niimi, H. Noda, and E. Kawaguchi. *An image embedding in image by a complexity based region segmentation method,* In *ICIP*, volume 3, pages 74–77, 1997.

[20] W.B. Pennebaker and J.L. Mitchell *JPEG Still Image Data Compression Standard,* Van Nostrand Reinhold, 1993.

[21] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn. *Attacks on Copyright Marking Systems,* in D. Aucsmith (Ed.), Information Hiding, LNCS 1525, IH'98, pp. 219-239 Springer, 1998.

[22] F.A.P. Petitcolas and R. J. Anderson. *Evaluation of Copyright Marking Systems,* in Proc. IEEE Multimedia Systems (ICMCS'99), vol. 1, pp. 574-579, June 1999, Florence, Italy.

[23] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn. *Information hiding – a survey,* in Proceedings of the IEEE, 87(7):1062–1078, July 1999.

[24] N. Provos. *Defending against statistical steganalysis,* in Proc. 10th USENIX Security Symposium , pages 323–335, August 2001.

[25] N. Provos. *Probabilistic methods for improving information hiding*, Technical Report 01-1, CITI, University of Michigan, January 2001.

[26] M. Ramkumar *Data Hiding in Multimedia — Theory and Applications,* PhD dissertation, Dept. of ECE, New Jersey Institute of Technology, Newark, NJ, 1999.

[27] M. Ramkumar, A.N. Akansu, and A.A. Alatan. *An FFT Based Signaling Scheme for Multimedia Steganography,* Preprint.

[28] C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication,* University of Illinois Press, 1949. Also appeared as a series of papers by Shannon in the Bell System Technical Journal, July 1948, October 1948 (A Mathematical Theory of Communication), January 1949 (Communication in the Presence of Noise).

[29] C.E. Shannon. *The Zero Error Capacity of a Noisy Channel,* IRE Trans. on Information Theory, Vol. IT-2, pp. S8-S19, Sept. 1956.

[30] G. Strang. *The discrete cosine transform,* SIAM Review, 41(1):135–147, 1999.

[31] G.J. Simmons. *The Prisoners' Problem and the Subliminal Channel,* D. Chaum (ed.) Advances In Cryptology: Proc. of Crypto 83, pp. 51-67, Plenum Press, 1984.

[32] G.J. Simmons. *the History of Subliminal Channels,* newblockIEEE J. on Selected Areas in Communications, v 16, no 4, pp. 452-273, April 1998.

[33] R. Sugihara. *Practical Capacity of Digital Watermarks,* in I.S. Moskowitz (Ed.) Information Hiding, LNCS 2137, IH 2001, pp. 316-329, Springer 2001.

[34] D. Upham. *Jpeg-Jsteg.* Modification of the Independent JPEG Group's JPEG software (release 4) for 1-bit steganography in JFIF output files, http://www.tiac.net/usres/lorejwa/jsteg.htm, 1997.

[35] B. R. Venkatraman and R. E. Newman-Wolfe *Capacity Estimation and Auditability of Network Covert Channels,* Proc. Symposium on Security and Privacy, Oakland, CA, pp. 186-198, IEEE, May 8-10, 1995.

[36] A. Westfeld. *F5 — A Steganographic Algorithm: High Capacity Despite Better Steganalysis,* in I.S. Moskowitz (Ed.) Information Hiding, LNCS 2137, IH 2001, pp. 289-302, Springer 2001.

[37] xv, http://www.trilon.com/xv/, 1994.